Kalman Filters (KF)

Content/figure credits go to
http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/

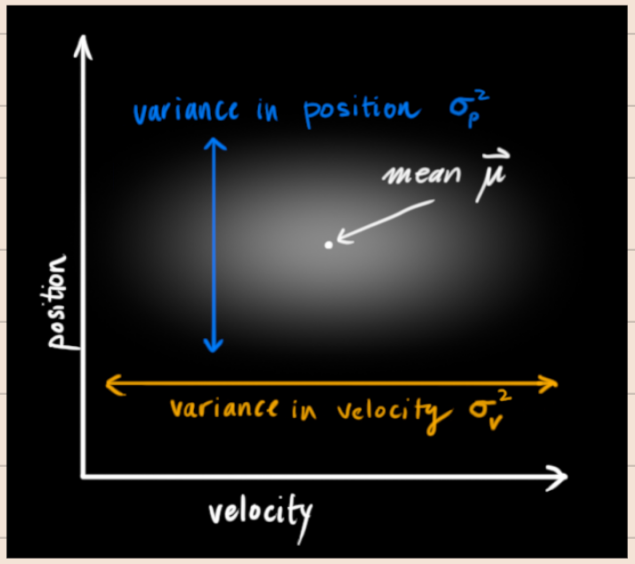
used to make an educated guess about a system's next state when we have uncertain information about a dynamic system

KF's in Robots

- state $\vec{x}_k = \begin{bmatrix} \vec{p} \\ \vec{v} \end{bmatrix}$ position / velocity
- there is GPS sensing available, but inaccurate ← "sensor"
- we know how the robot should move (its controls) but we can't deduce the robot's next state perfectly due to e.g. wheel slippage, wind, etc.
  ↑ "prediction"

* sensor + prediction = better info about robot's next state?

How do KF's see this problem?



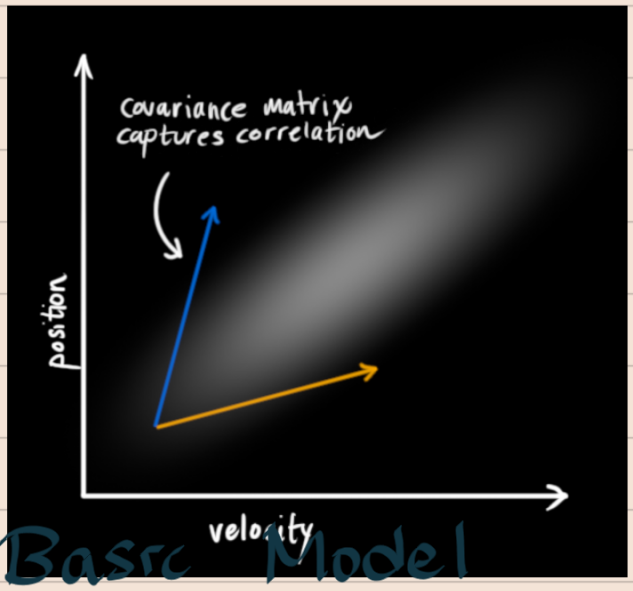
assumes $\vec{p}$ and $\vec{v}$ are Gaussian distributed (some more likely than others)

sometimes, $\vec{v}$ and $\vec{p}$ are correlated:

⇒ $\vec{v}$ tells us something about $\vec{p}$ and vice versa

⇒ eg. estimating a robot's new $\vec{p}$ based on its previous $\vec{v}$

covariance matrices Σ capture correlation



Basic Model

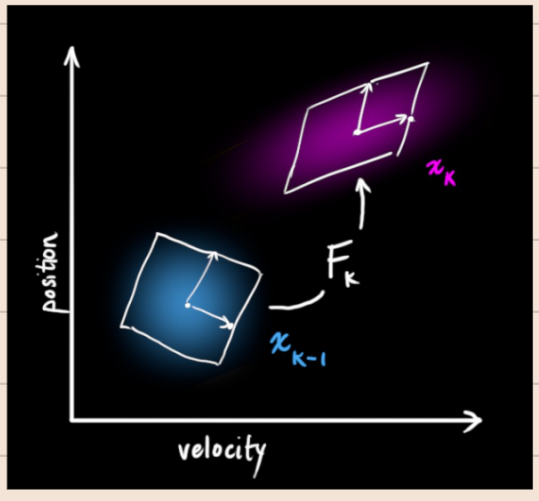
$$\begin{cases} \text{best estimate } \hat{x}_k = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix} & \text{"mean"} \\ \text{covariance matrix } P_k = \begin{bmatrix} \Sigma_{pp} & \Sigma_{pv} \\ \Sigma_{vp} & \Sigma_{vv} \end{bmatrix} \\ \text{prediction matrix } F_k \quad \text{updates } \hat{x}_{k-1} \rightarrow \hat{x}_k \end{cases}$$

Ex) $p_k = p_{k-1} + \Delta t\, v_{k-1}$

$v_k = v_{k-1}$

$$\Rightarrow \hat{x}_k = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{F_k} \hat{x}_{k-1}$$

this maps ALL points from the previous timestep to the next!



how do we update covariance, P_k?

It turns out: $\text{Cov}(x) = \Sigma \Rightarrow \text{Cov}(Ax) = A\Sigma A^T$

Thus: since $\hat{x}_k = F_k \hat{x}_{k-1}$ and $\text{Cov}(\hat{x}_{k-1}) = P_{k-1}$,

$$\begin{aligned}\text{Cov}(\hat{x}_k) &= \text{cov}(F_k \hat{x}_{k-1}) \\ &= F_k P_{k-1} F_k^T\end{aligned}$$

In summary, we can update $\hat{x}_k$ and $\hat{P}_k$ according to:

$$\boxed{\begin{aligned}\hat{x}_k &= F_k \hat{x}_{k-1} \\ P_k &= F_k P_{k-1} F_k^T\end{aligned}}$$

Adding external influence (controls)

$\hat{u}_k$: control vector / matrix

Ex) $p_k = p_{k-1} + \Delta t\, v_{k-1} + \frac{1}{2} a \Delta t^2$ — a: acceleration due to throttle setting

$v_k = v_{k-1} + a\Delta t$

$$\Rightarrow \hat{x}_k = F_k \hat{x}_{k-1} + \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix} a$$

$$= F_k \hat{x}_{k-1} + B_k \vec{u}_k$$

External uncertainty

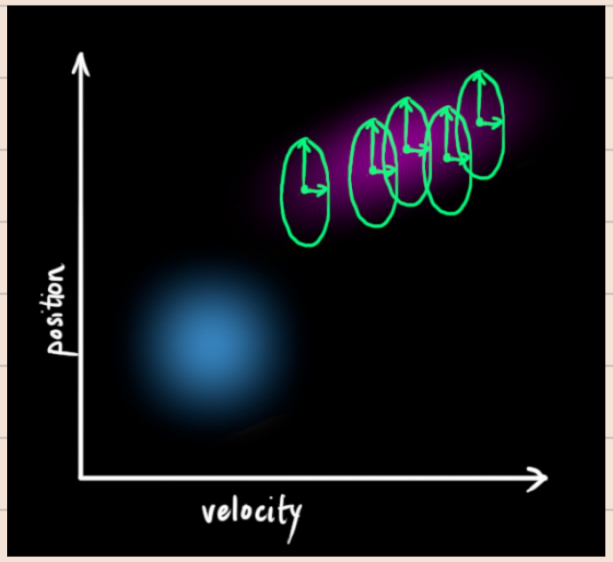
models forces / influences we don't know about

$$\hat{x}_k = F_k \hat{x}_{k-1} + B_k u_k$$

$$P_k = F_k P_{k-1} F_k^T + Q_k$$

(noise)

add this term to our prediction step



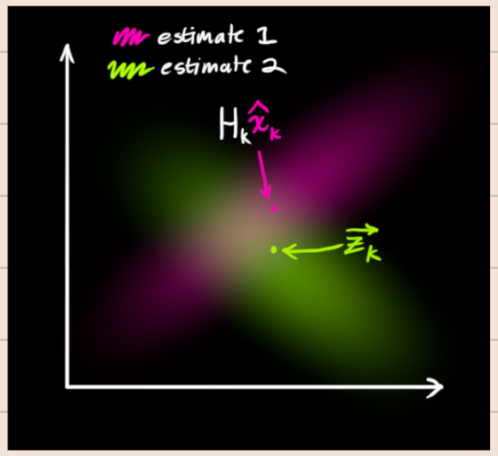
Refining w measurements (sensors)

H_k : sensor model
(the units and scale of sensor reading may not be the same as the state)

$$\begin{cases} \vec{\mu}_{\text{expected}} = H_k \hat{x}_k \\ \Sigma_{\text{expected}} = H_k P_k H_k^T \end{cases}$$

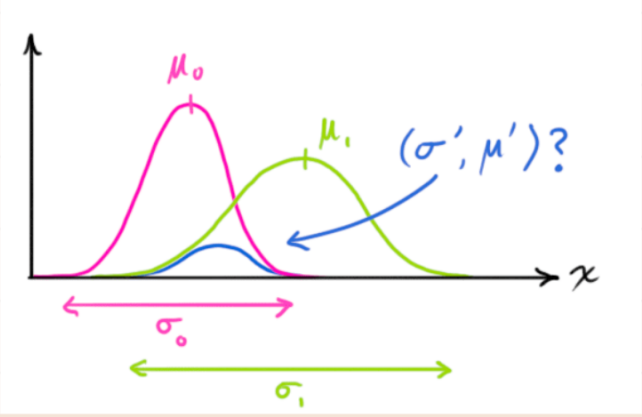
R_k : covariance / sensor noise
z_k : sensor reading aka "mean"

* we now have 2 Gaussians to represent the robot's state:
  ① prediction $\hat{x}_k$
  ② sensor reading z_k



what's the most likely state?
⇒ multiply the two gaussians together
⇒ product gives a new gaussian!

Combining 2 Gaussians



1D (scalar version)

$$\begin{cases} \mu' = \mu_0 + k(\mu_1 - \mu_0) \\ \sigma'^2 = \sigma_0^2 - k\sigma_0^2 \\ \text{where } k := \dfrac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2} \end{cases}$$

multi-dimensional (matrix)

$$* \begin{cases} \vec{\mu}' = \vec{\mu}_0 + K(\vec{\mu}_1 - \vec{\mu}_0) \\ * \quad \Sigma' = \Sigma_0 - K\Sigma_0 \end{cases}$$

where $K = \Sigma_0(\Sigma_0 + \Sigma_1)^{-1}$

"Kalman gain"

Summary of KF

We have 2 distributions

① predicted measurement : $(\mu_0, \Sigma_0) = (H_k \hat{X}_k, H_k P_k H_k^T)$

② observed measurement (sensor) : $(\mu_1, \Sigma_1) = (\vec{z}_k, R_k)$

Combining the 2 Gaussians, we have the following refined prediction about the robot's state:

new best estimate from combining ① & ②

$$* \quad H_k X_k' = H_k \hat{X}_k + K(\vec{z}_k - H_k \hat{X}_k) \qquad \text{state (mean)}$$

$$* \quad H_k P_k' H_k^T = H_k P_k H_k^T - K H_k P_k H_k^T \qquad \text{covariance}$$

where the Kalman gain K is:

$$K = H_k P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}$$